Serdes Introduction and options for Macro-modelling

Richard Ward
11-Apr-2006

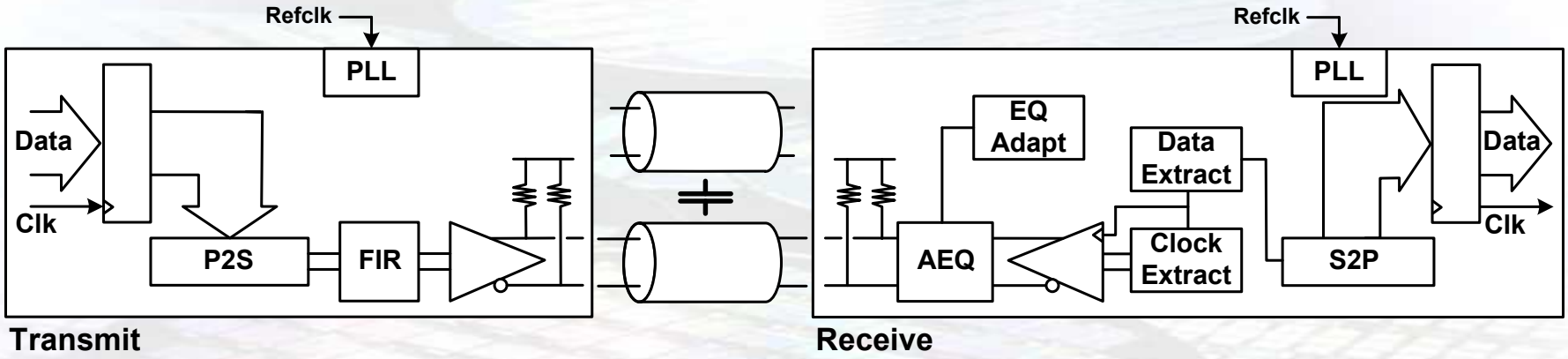
Expectation

- **There are as many opinions on SerDes as there are SerDes available.**
  - The following gives a basic overview of some SerDes components, mostly found on 1-6.25Gbps devices. Higher rate architectures could be discussed later.

Standards for SerDes?

- **Which Standard would you like to follow?**
  - 802.3ap (10G Ethernet over backplane)
  - OIF CEI 6G (short and long reach)
  - OIF CEI 11G (short and long reach)
  - PCI-Express (2.5G gen1, 5G gen2)
  - XAUI (3.125G)
  - Serial ATA (3.125G)
  - FBDIMM (4.8/6.4/9.6G)
  - SerialRapidIO, (1.25/2.5/3.125G)
  - Infiniband
  - FibreChannel
  - …

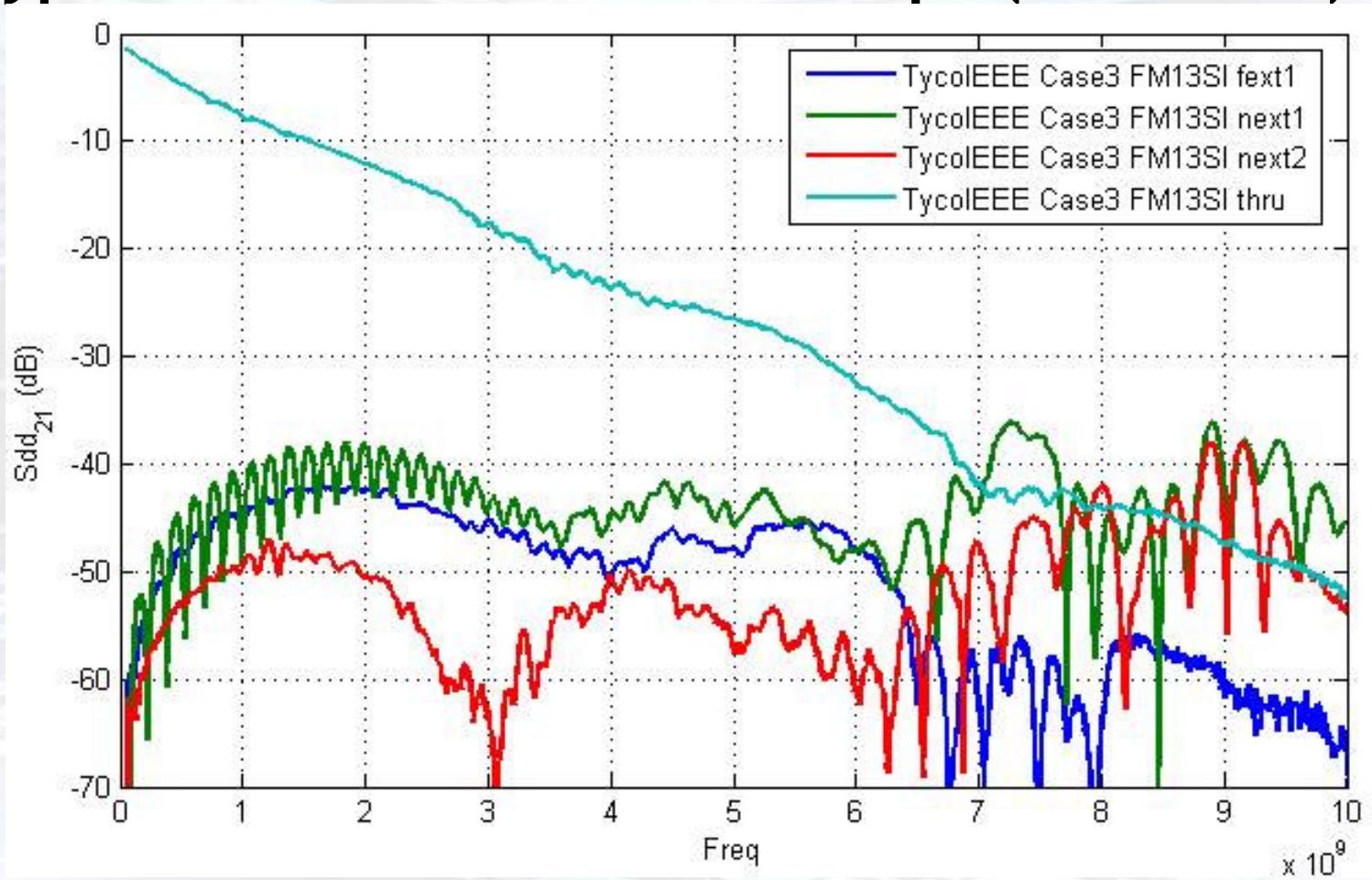
Typical SerDes Block Diagram



- **Differential data (usually CML) has Clock and Data embedded into a serial stream. Most are NRZ/binary.**
- **Keys parameters are:**
  - Jitter generation (Tx) and jitter tolerance (Rx)
  - Equalization ability (both Tx and Rx sides) and adaption
  - Asynchronous tracking rate of Rx
  - PLL jitter rejection (from Refclk and chip/system noise)
- **Most higher rate systems are moving to Rx DFE (analog or digital) and possibly PAM4.**

Example Channel

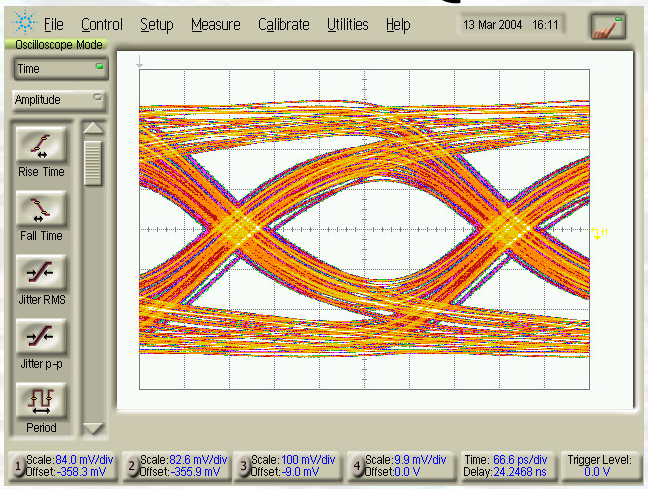
- Typical channel for 5 to 10Gbps (~40” FR4)



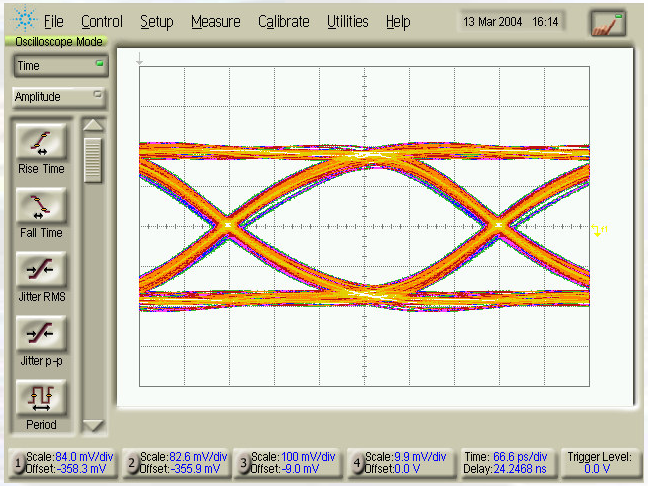
So what do the signals look like?

- 3.125Gbps

No EQ

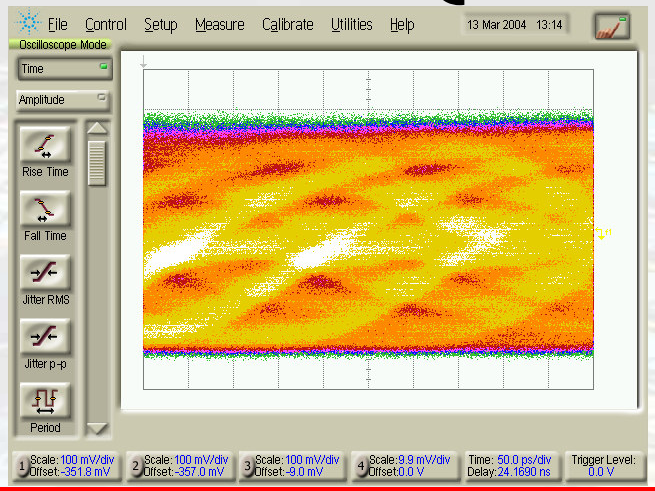


Tx FIR

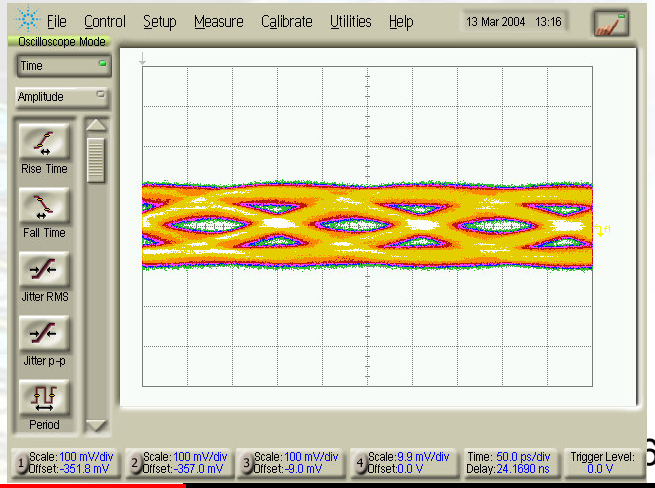


- 6.25Gbps

No EQ



Tx FIR



~40” FR4 o/p

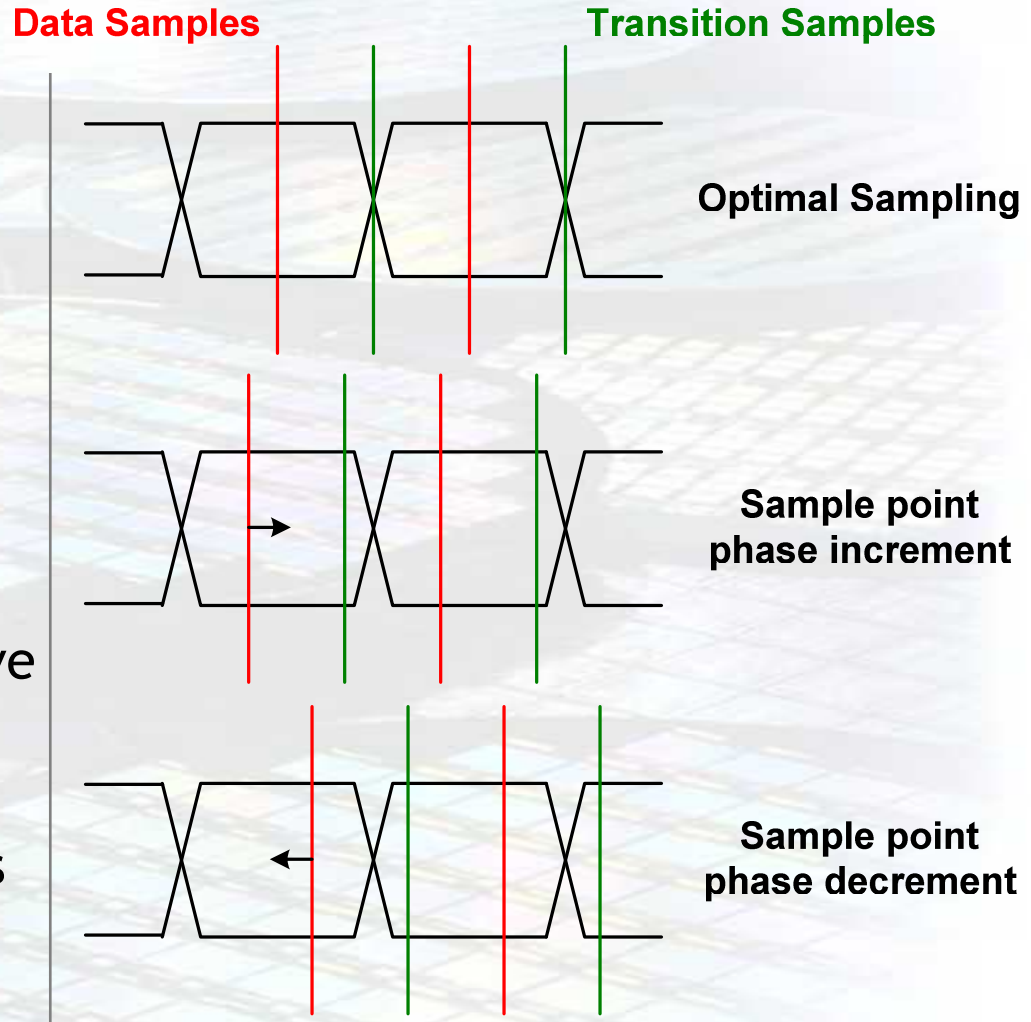
Typical Components: Tx FIR

- **Transmit FIR, 4 or 5 taps is common**
  - Most Tx FIR systems are not adaptable today (should arrive shortly).
  - 802.3ap proposes a “backchannel” communication path for the Tx adaption.
  - Most proposals for backchannels are:
    - Slower than the main data
    - Either:
      - On the same wires (such as adjusting common mode)

      or
      - Have a separate wire-link for the adaption (1 backchannel link may send adaption information for many Data links).

Simple Clock and Data Recovery I

- **A common algorithm:**
  - Over-sample the data
  - Track the “transitions” while having a 90 deg offset for data sampling
  - Sample point will “hunt” and introduce an effective jitter.
  - Relies on data transitions



Simple Clock and Data Recovery II

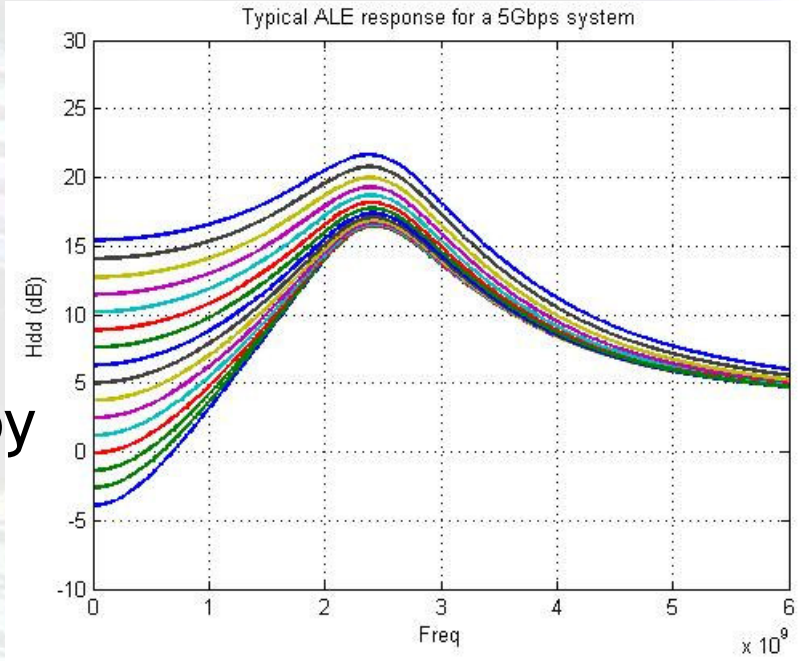
- **The algorithms are usually written in verilog/vhdl and compare data and transition values across edges to determine phase incr/decr required.**
- **Parametric/functional options are commonly:**
  - voting window and update time adjustment
  - threshold for incr/decr of phase (based on vote accumulation)
  - variables phase rate updates
    (S-ATA requires the ability to track +/- 5000ppm, many CPU/switch applications require +/-100ppm)
  - ability for FASTLOCK on startup or sync loss
  - ability to LOCK or STEP phases (primarily test features)
- **Not really suited to a macro model due to algorithm complexity and that the code usually exists in a standard AMS sub-language.**

Receiver Equalisation I

- **2 main camps for RX EQ:**
  - Analog Linear EQ (ALE) [mostly <6Gbps]
  - Decision Feedback EQ (DFE) [mostly >6Gbps]

Simple ALEs will adapt
by adjusting dc vs ac gain

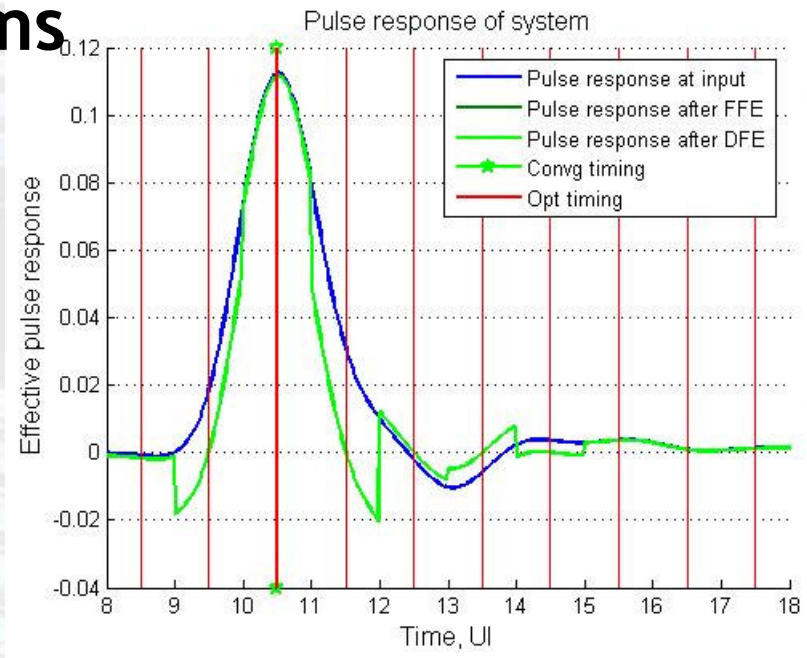
Adaption algorithms vote
on incr/decr sample controls by
knowing the pattern received
and the expected ISI from that
pattern



Receiver Equalisation II

- DFEs have more advanced EQ performance
- Current systems calculate up to 10 taps
- Floating taps for echo cancellation are possible
- Calculations can be shared between CDR and EQ depending on algorithms

- Example pulse response shows correction in the time domain where the coefficients cause the 5 main (non-cursor) taps to be zeroed.



IBIS/AMS Macro models? Discussion

- **Possible models for SerDes components:**
  - Differential, clocked samplers/comparators with slew based offsets
  - Gain/Attenuation stages with pole/zero or tables
  - PLL, including noise/jitter modelling
    (Difficult trade-off of Execution Speed vs Functionality)
  - Transmit driver with programmable FIR and return loss tables? (Too big a block?)
  - Noise/jitter injection from just about everything!
- **Unlikely to model CDR or EQ as components, embedded code seems a better solution.**
- **Concerns:**
  - Small blocks may be too fine grain to be useful
  - Large blocks become individual and functionally complex (and still unlikely to match many real SerDes)